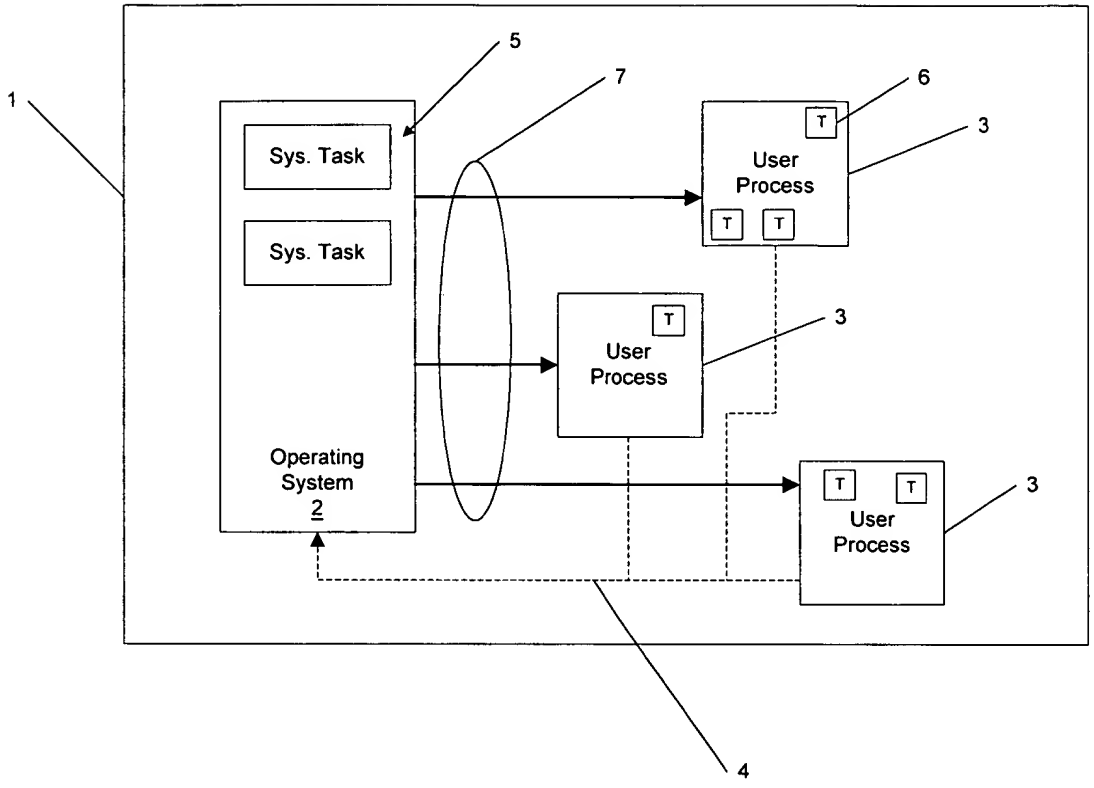


Figure 1
(Prior Art)

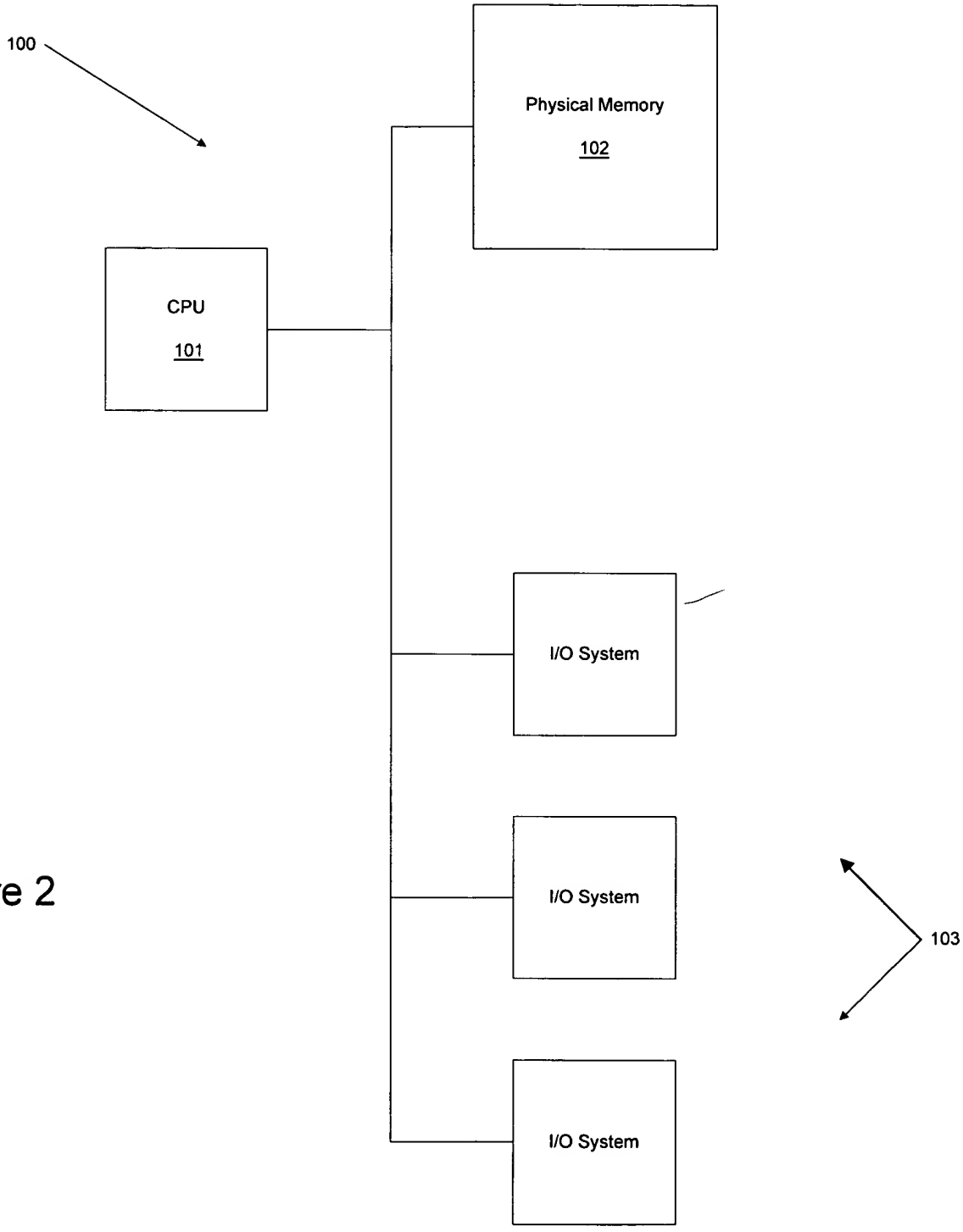

100
Physical Memory
102
CPU
101
I/O System
I/O System
I/O System
103

Figure 2

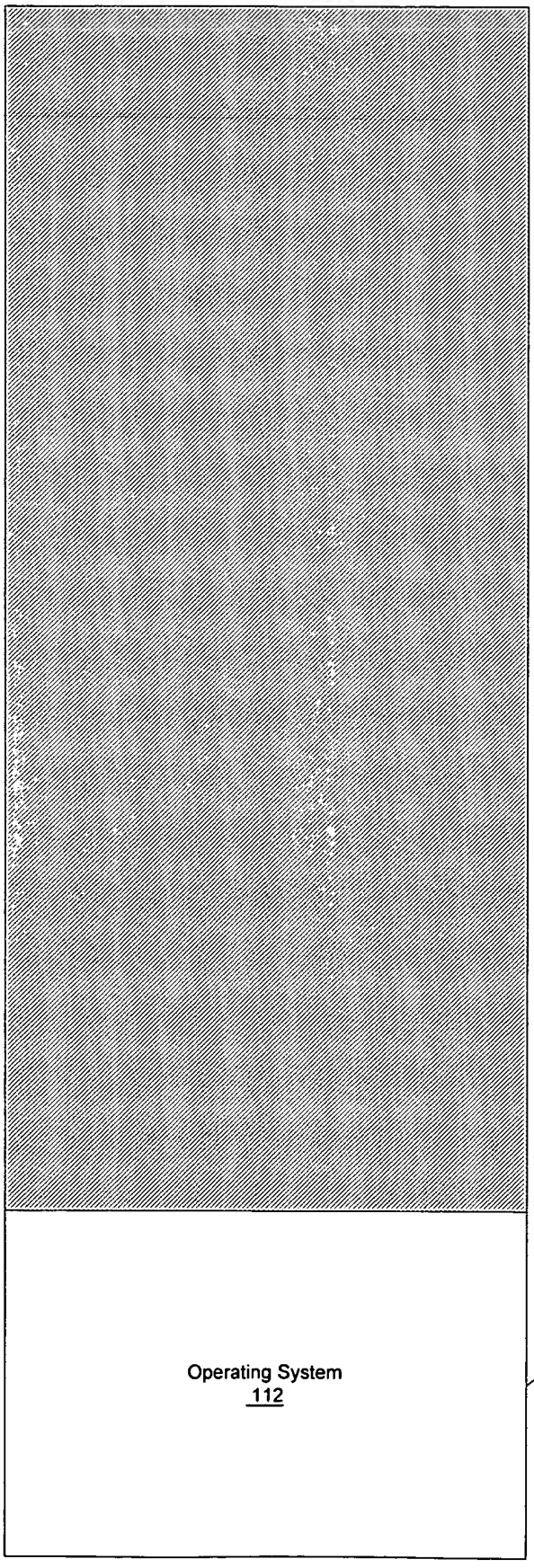


Figure 3

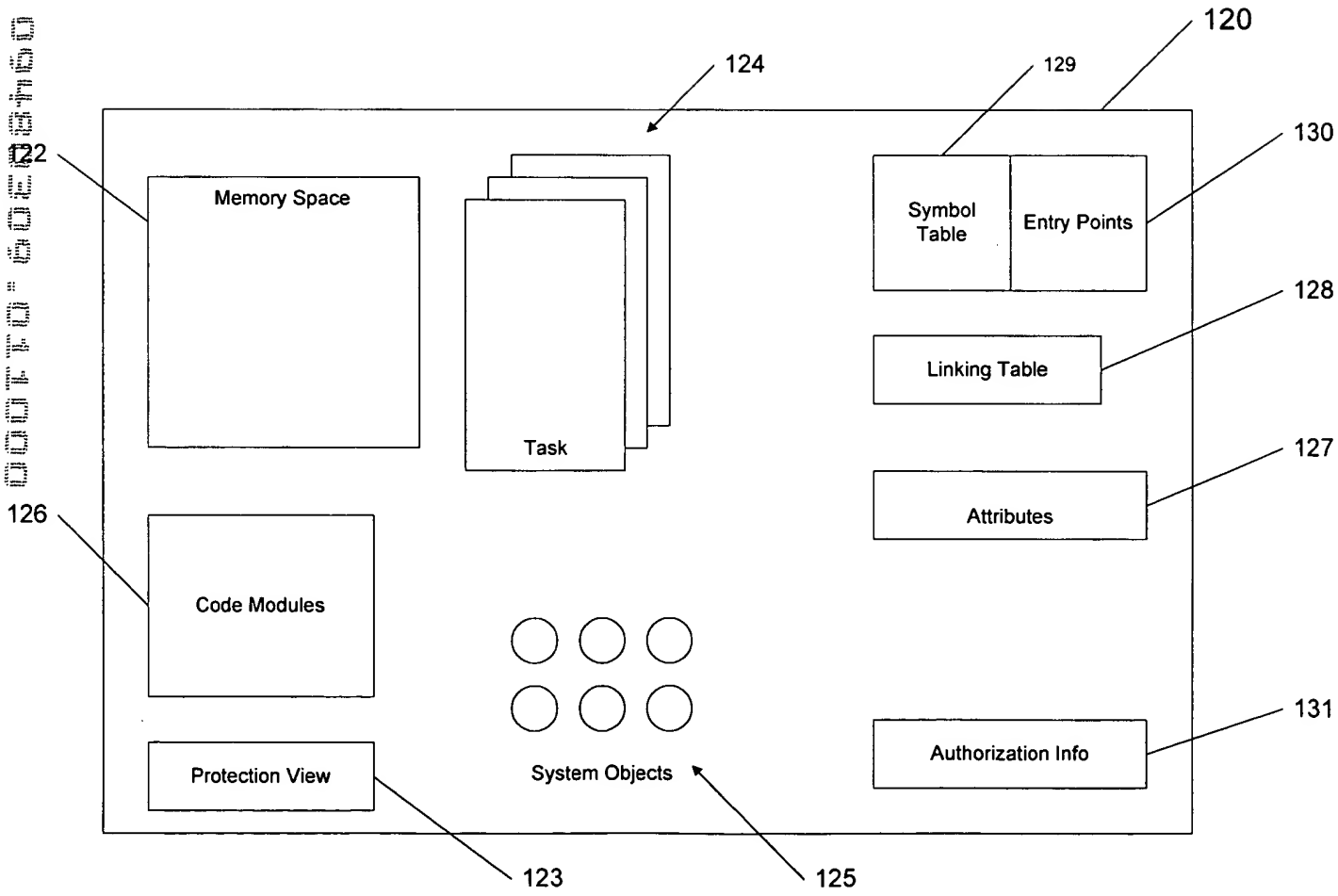

120
124
129
122
130
Memory Space
Task
Symbol Table
Entry Points
128
Linking Table
127
Attributes
126
Code Modules
System Objects
131
Authorization Info
Protection View
123
125

Figure 4

Figure 5

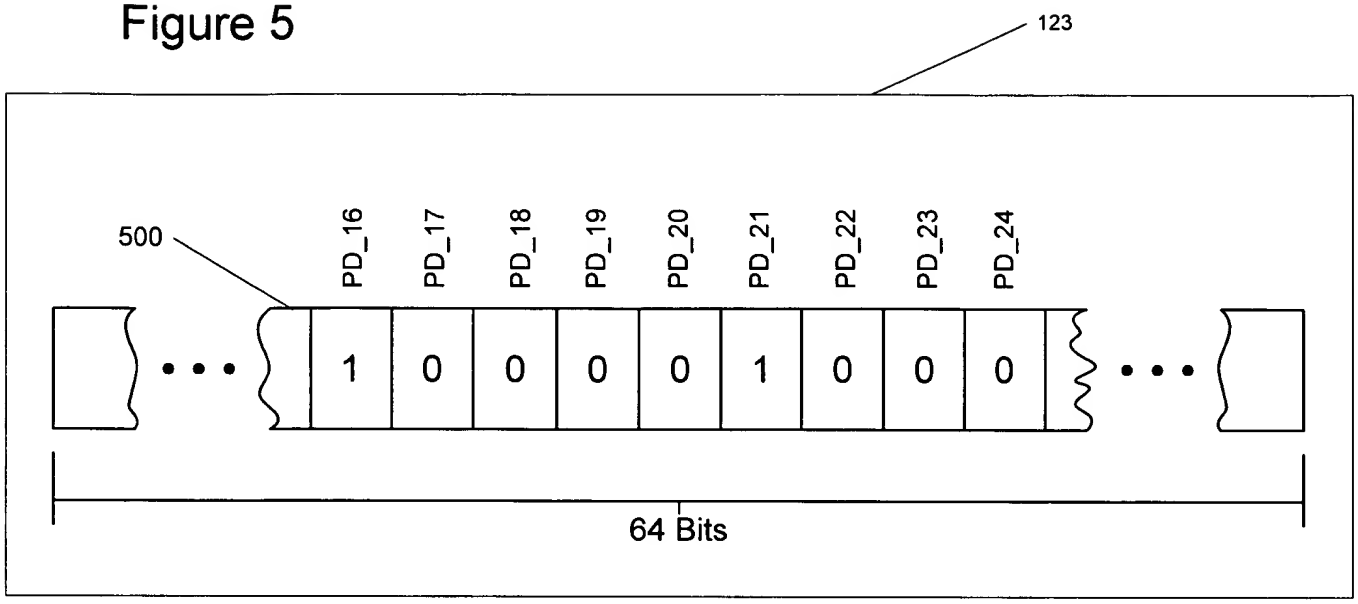
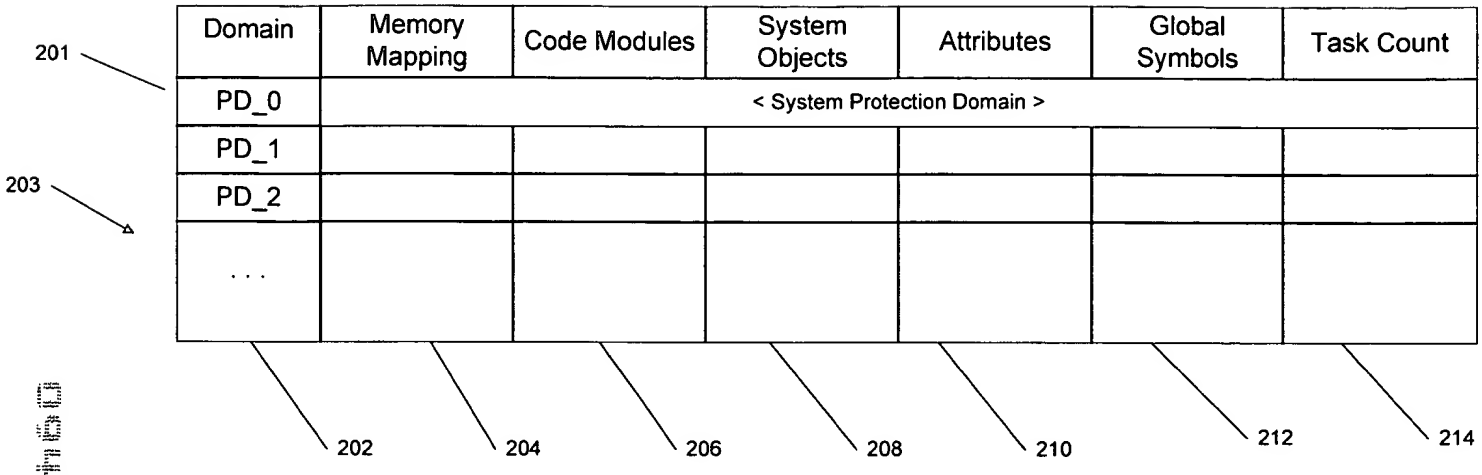

123
500
PD_16
PD_17
PD_18
PD_19
PD_20
PD_21
PD_22
PD_23
PD_24
1
0
0
0
0
1
0
0
0
64 Bits

Figure 6



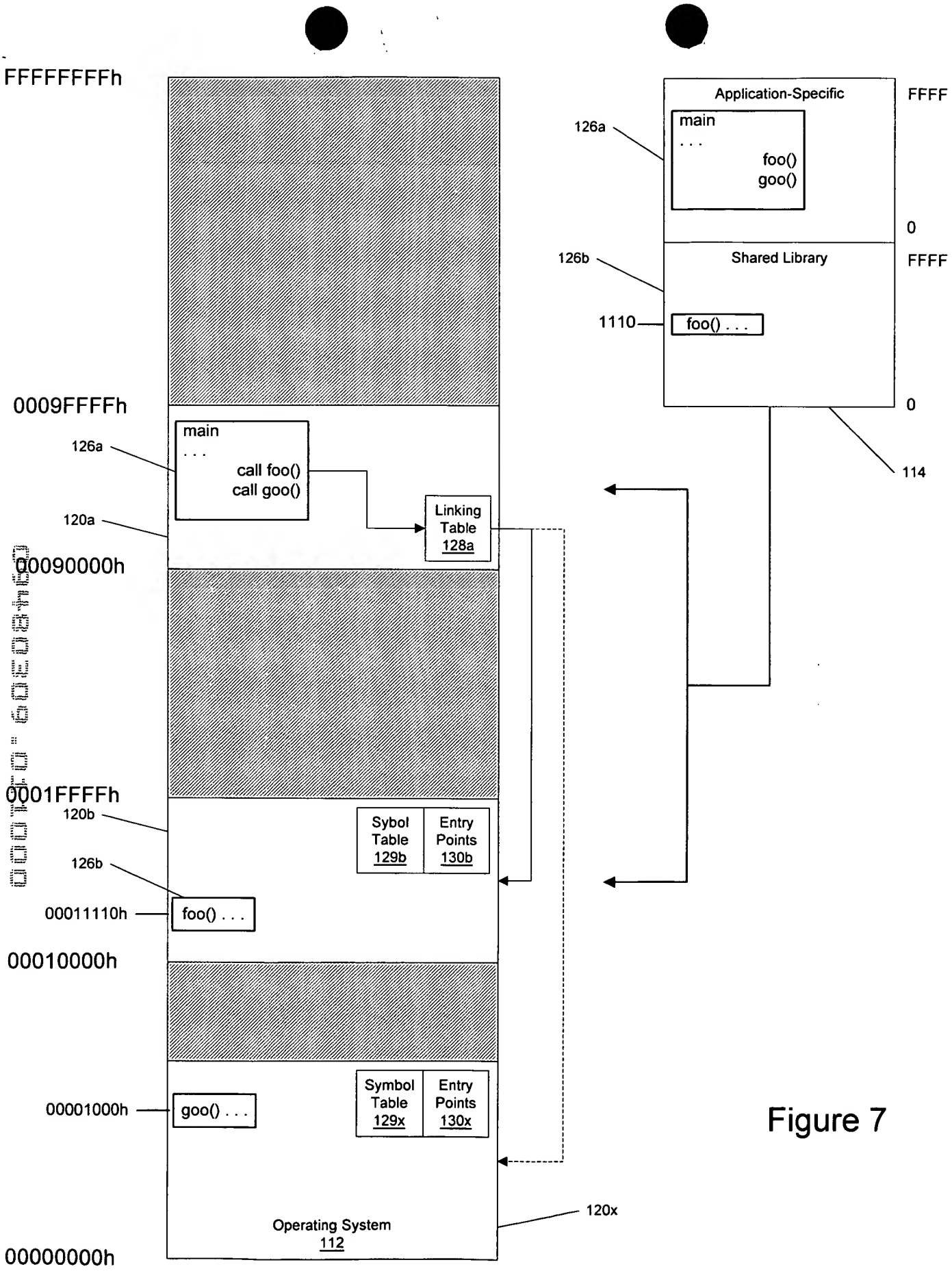

FFFFFFFFh
0009FFFFh
00090000h
0001FFFFh
00010000h
00001000h
00000000h
Application-Specific
FFFF
main
. . .
foo()
goo()
126a
0
126b
Shared Library
FFFF
1110
foo() . . .
0
114
main
. . .
call foo()
call goo()
126a
120a
Linking Table 128a
Sybol Table 129b
Entry Points 130b
120b
126b
00011110h
foo() . . .
Symbol Table 129x
Entry Points 130x
00001000h
goo() . . .
Operating System 112
120x

Figure 7

Figure 8

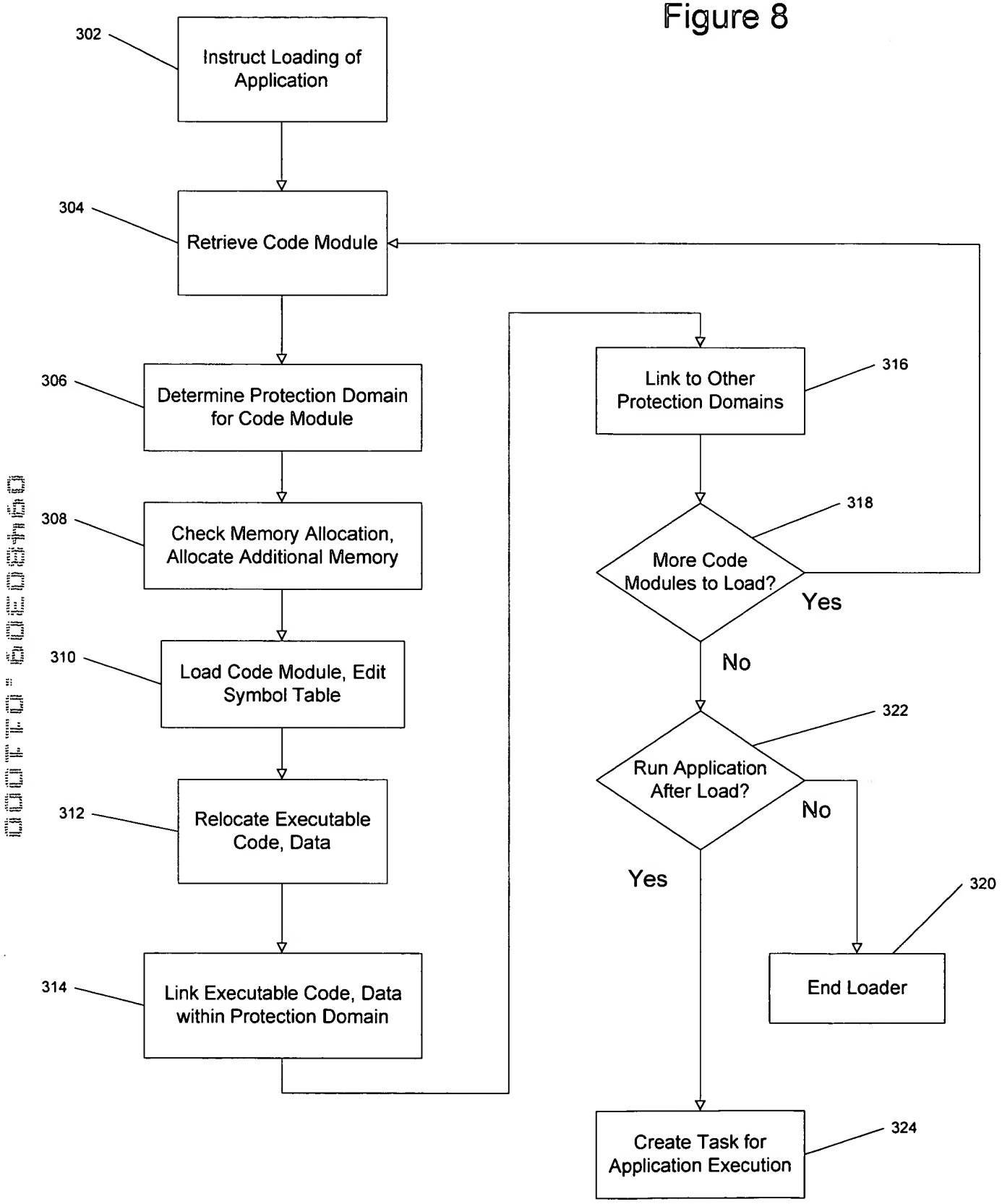

302
Instruct Loading of Application
304
Retrieve Code Module
306
Determine Protection Domain for Code Module
308
Check Memory Allocation, Allocate Additional Memory
310
Load Code Module, Edit Symbol Table
312
Relocate Executable Code, Data
314
Link Executable Code, Data within Protection Domain
316
Link to Other Protection Domains
318
More Code Modules to Load?
Yes
No
322
Run Application After Load?
No
Yes
320
End Loader
324
Create Task for Application Execution

Figure 9

129

Entry #	Symbol	Symbol Address	Entry Point?
0	<symbol_name_0>	<address_0>	< y/n >
1	<symbol_name_1>	<address_1>	< y/n >
2	<symbol_name_2>	<address_2>	< y/n >
3	<symbol_name_3>	<address_3>	< y/n >
4	<symbol_name_4>	<address_4>	< y/n >
. . .	. . .	. . .	. . .

Figure 13

129b

Entry #	Symbol	Symbol Address	Entry Point?
. . .	. . .	. . .	. . .
4	foo	00011111	Y
. . .	. . .	. . .	. . .

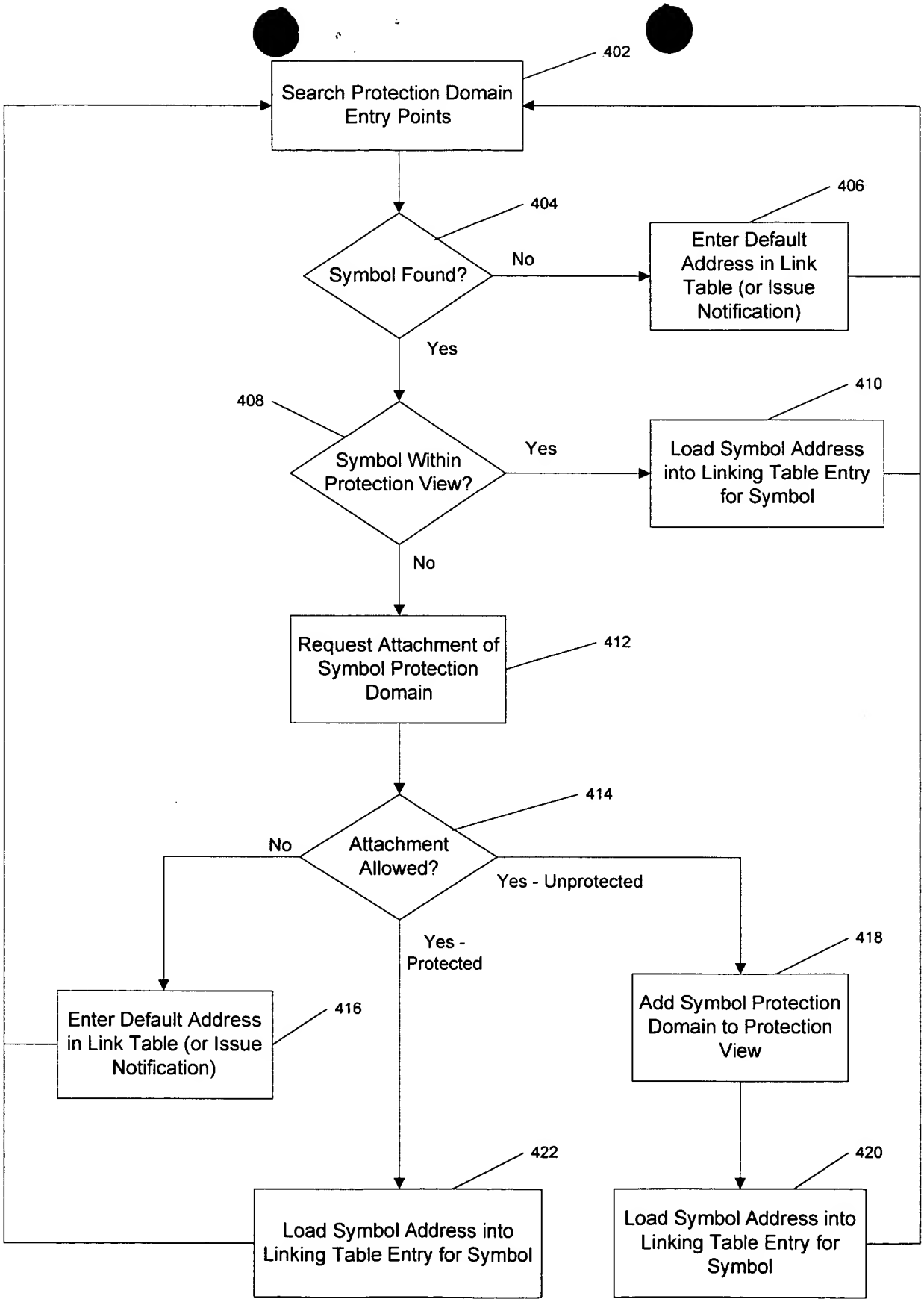

Search Protection Domain Entry Points
402
Symbol Found?
404
No
Enter Default Address in Link Table (or Issue Notification)
406
Yes
Symbol Within Protection View?
408
Yes
Load Symbol Address into Linking Table Entry for Symbol
410
No
Request Attachment of Symbol Protection Domain
412
Attachment Allowed?
414
No
Yes - Unprotected
Yes - Protected
Enter Default Address in Link Table (or Issue Notification)
416
Add Symbol Protection Domain to Protection View
418
Load Symbol Address into Linking Table Entry for Symbol
422
Load Symbol Address into Linking Table Entry for Symbol
420

Figure 10

Figure 11

128

132 | 133

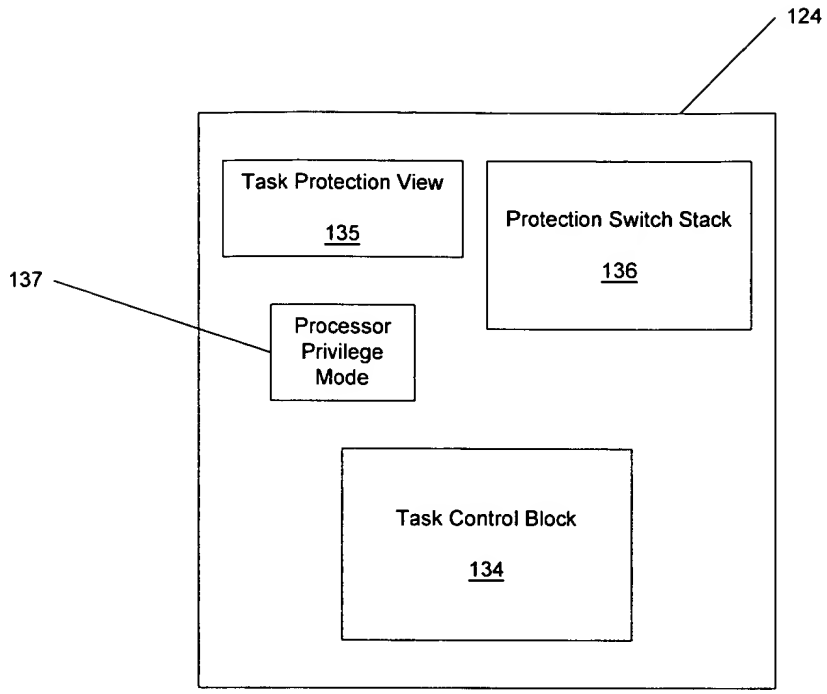
Entry #	Symbol	Link Stub
0	<symbol_name_0>	JMP <address_0>
1	<symbol_name_1>	JMP <address_1>
2	<symbol_name_2>	JMP <address_2>
3	<symbol_name_3>	JMP <address_3>
4	<symbol_name_4>	JMP <address_4>
. . .	. . .	. . .

Figure 12

128a

Entry #	Symbol	Link Stub
0	foo	JMP 00011110
1	goo	JMP 00001000

Figure 14


124
Task Protection View
135
Protection Switch Stack
136
137
Processor
Privilege
Mode
Task Control Block
134

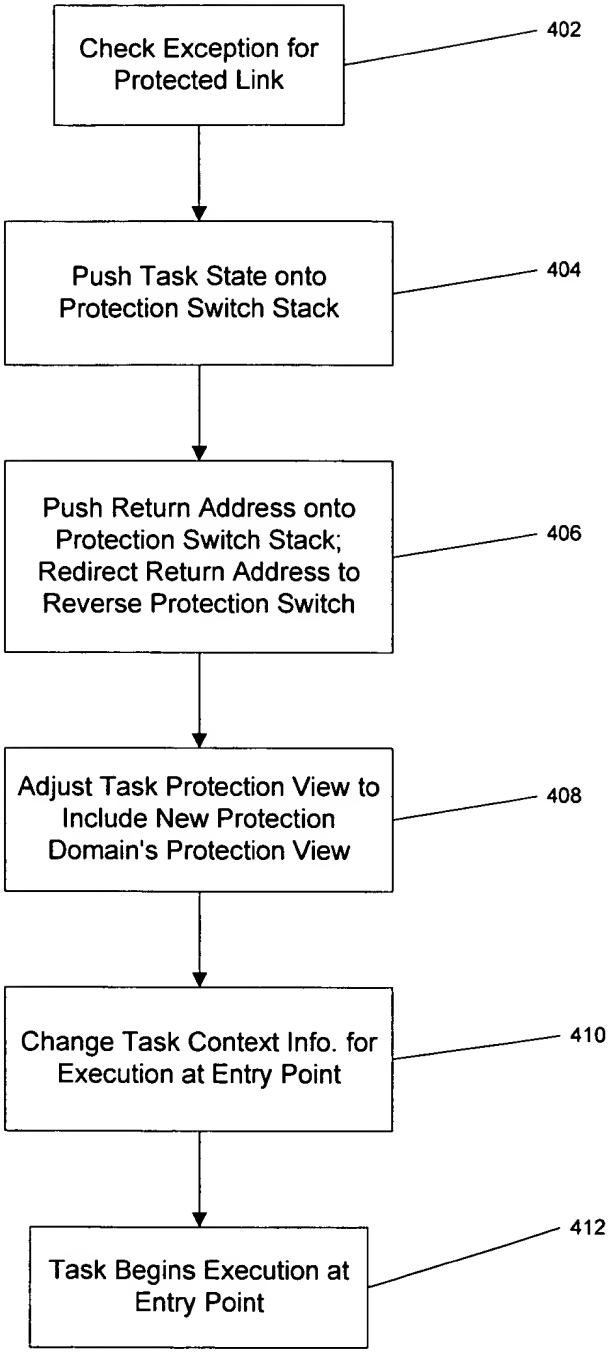
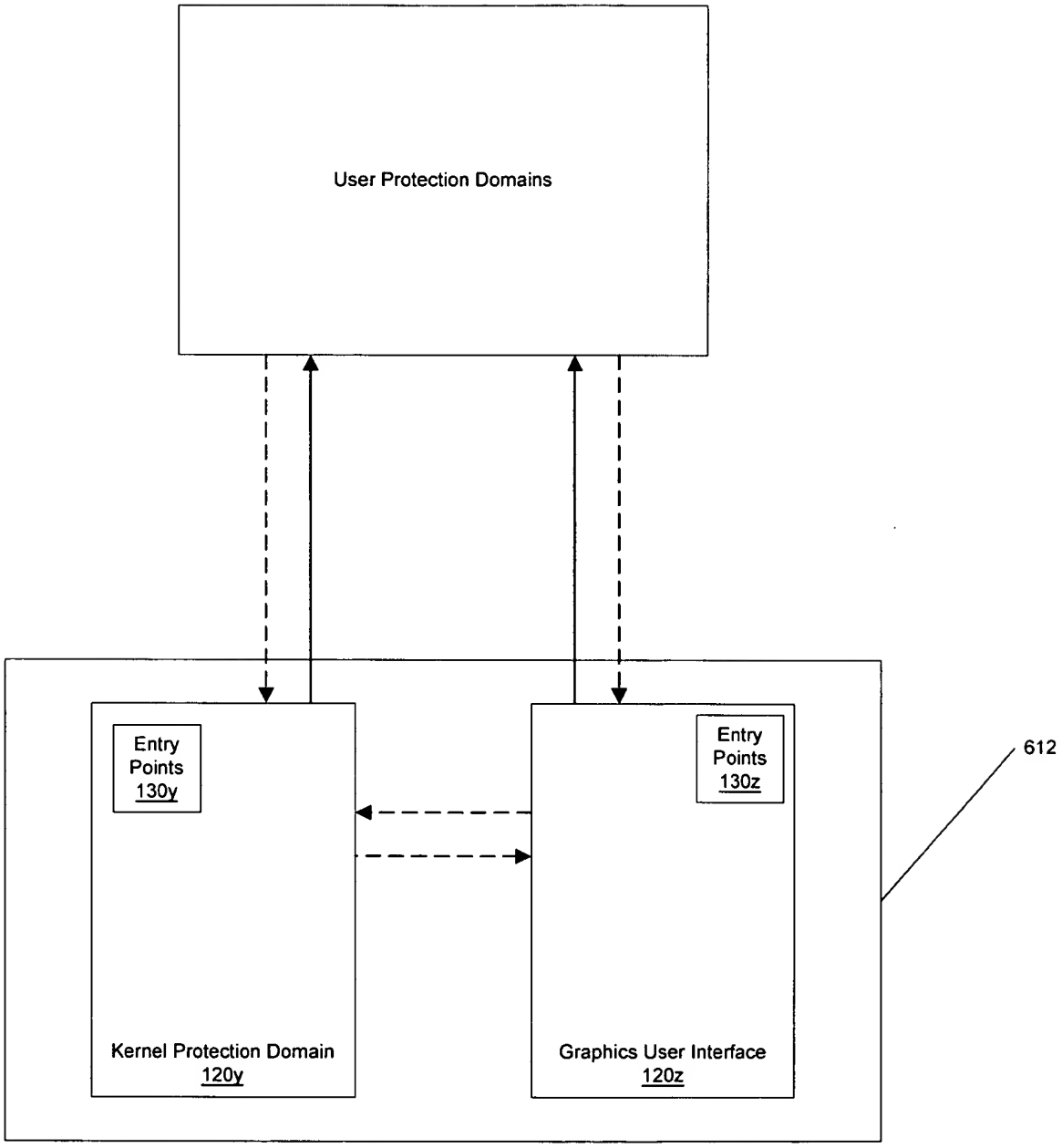

Check Exception for Protected Link
402
Push Task State onto Protection Switch Stack
404
Push Return Address onto Protection Switch Stack; Redirect Return Address to Reverse Protection Switch
406
Adjust Task Protection View to Include New Protection Domain's Protection View
408
Change Task Context Info. for Execution at Entry Point
410
Task Begins Execution at Entry Point
412

Figure 15

Figure 16


User Protection Domains
Entry Points 130y
Entry Points 130z
612
Kernel Protection Domain 120y
Graphics User Interface 120z

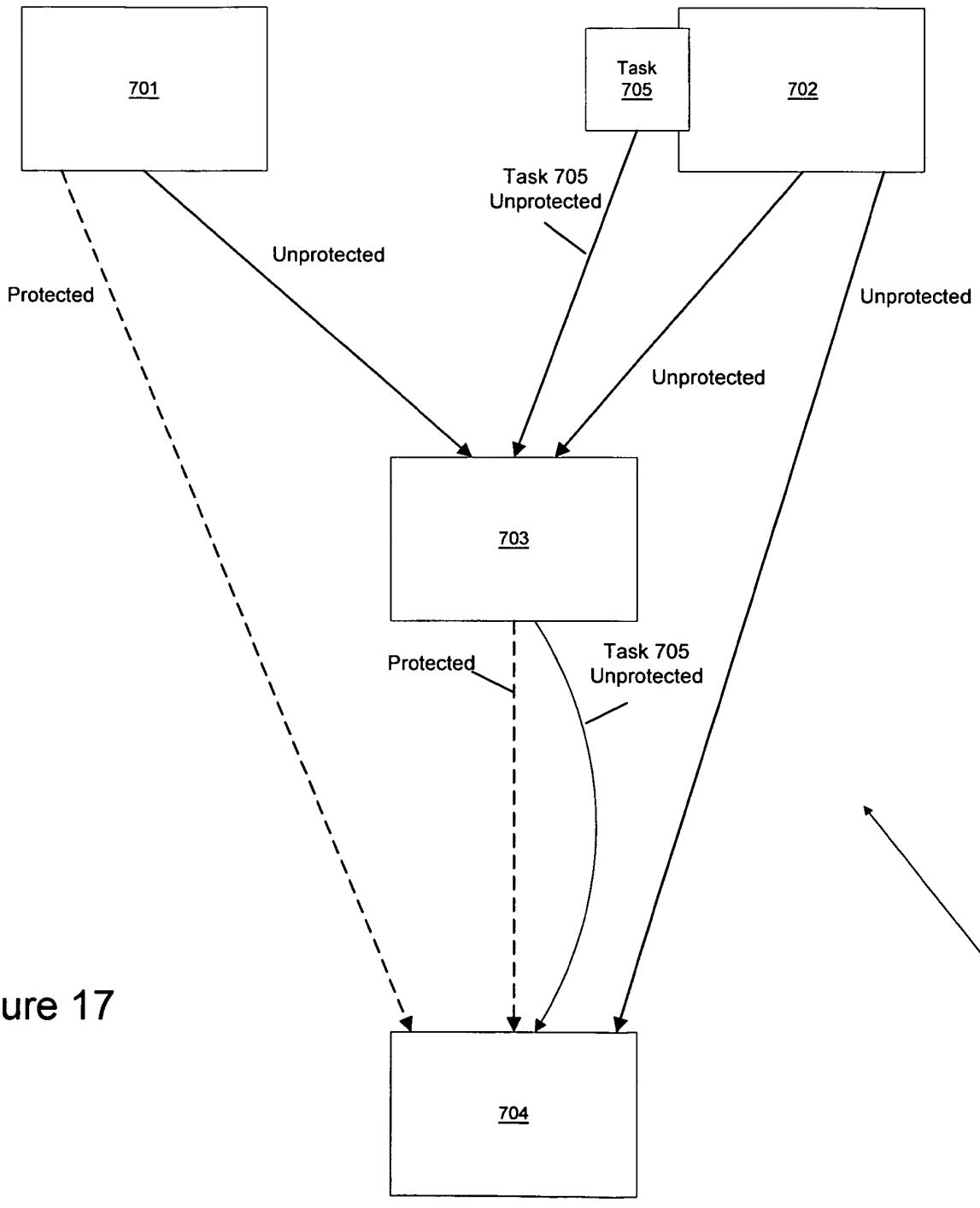

701
Task
705
702
Task 705
Unprotected
Unprotected
Protected
Unprotected
Unprotected
703
Protected
Task 705
Unprotected
704
700

Figure 17

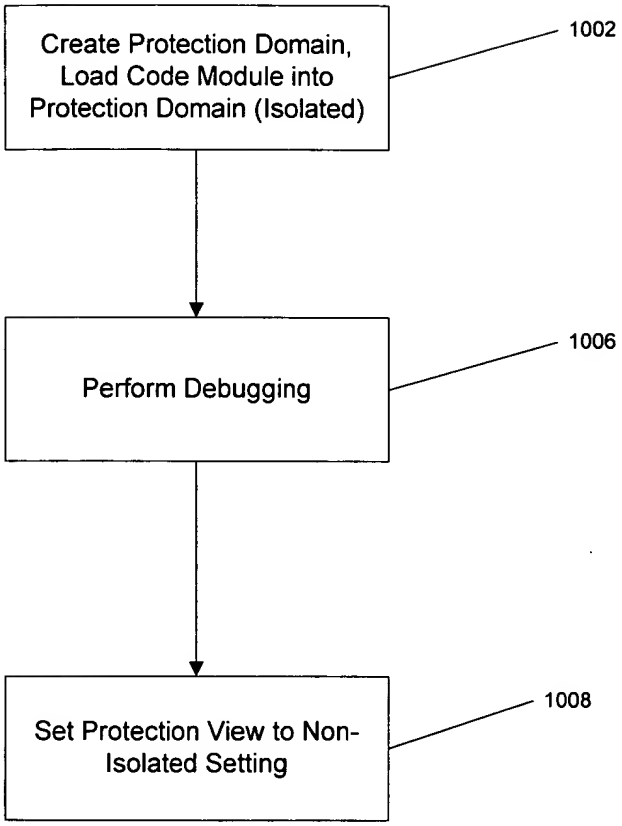

Create Protection Domain, Load Code Module into Protection Domain (Isolated)
1002
Perform Debugging
1006
Set Protection View to Non-Isolated Setting
1008

Figure 18